

7.8

11/84

505-37-03-02  
ICASE COMPUTER SCIENCE PROGRAM

- NASA AND ICASE HAVE LARGE INVESTMENT IN SCIENTIFIC COMPUTING
- FUTURE OF SCIENTIFIC COMPUTING DEPENDENT UPON EFFECTIVE UTILIZATION OF PARALLEL COMPUTERS
- PARALLEL COMPUTING IS SYSTEM ISSUE REQUIRING INTERACTION OF:
  - PROBLEM DECOMPOSITION
  - ALGORITHM DEVELOPMENT
  - PROGRAMMING ENVIRONMENTS
  - COMMUNICATION & SYNCHRONIZATION
  - ARCHITECTURE

PRECEDING PAGE BLANK, NOT FILMED

## PROBLEM DECOMPOSITION

### MAP ALGORITHM ONTO AN ARRAY OF PROCESSORS

- BALANCE COMPUTATIONAL LOAD
- REDUCE COMMUNICATION
- REDUCE SYNCHRONIZATION

### DEVELOPING MAPPING HEURISTICS FOR CLASSES OF COMPUTATIONAL GRAPHS

EXAMPLE: MAP TREE ONTO MESH

## ALGORITHM DEVELOPMENT

### EXTENSIVE REVIEW OF NUMERICAL METHODS FOR PDES ASYNCHRONOUS ALGORITHMS

- ITERATIVE METHODS ON SIMULATOR AND FLEX
- NEW ALGORITHM FOR TIME DEPENDENT PROBLEMS

### ADAPTIVE METHOD FOR TIME DEPENDENT PROBLEMS

- BALANCE PROCESSOR LOAD
- REDUCE COMMUNICATION

PROGRAMMING ENVIRONMENTS

PISCES - PARALLEL IMPLEMENTATION OF SCIENTIFIC  
PROGRAMMING ENVIRONMENTS

FORTAN - UNIX BASED

PROVIDE USERS WITH DIFFERENT VIEW OF SYSTEM  
DEPENDING ON INTERESTS AND NEEDS

ENHANCE PORTABILITY OF APPLICATIONS PROGRAMS

BLAZE

NEW PARALLEL SCIENTIFIC PROGRAMMING LANGUAGE

OFFERS ADVANCED SUPPORT FOR ARRAY COMPUTATIONS

SIMPLIFIES EXTRACTION OF PARALLELISM BY COMPILER

COMMUNICATION AND SYNCHRONIZATION (C. & S.)

MAJOR OVERHEAD ASSOCIATED WITH PARALLEL COMPUTATION

LACK FORMAL METHODOLOGY FOR STUDY SUCH AS COMPUTATIONAL  
COMPLEXITY FOR NUMERICAL ALGORITHMS

DEVELOPED TWO MODELS THAT RELATE C. & S. COSTS TO:

ALGORITHM CHARACTERISTICS  
PROBLEM SIZE

SEEK GENERIC CHARACTERISTICS OF C. & S. THAT IMPACT:

PROBLEM DECOMPOSITION  
ALGORITHM DEVELOPMENT  
ARCHITECTURE

VALIDATE WITH ACTUAL COSTS ON REAL PARALLEL SYSTEMS

PARALLEL ARCHITECTURES

EVALUATION OF ARCHITECTURAL FEATURES IN VIEW OF SPECIFIC ALGORITHMS

- BIN PACKING PROBLEM
- MULTIGRID ALGORITHM
- SPARSE MATRIX ITERATION
- ADAPTIVE ALGORITHM

TOOLS

- SIMON SIMULATOR
- FLEX WITH DIFFERENT LOCAL/GLOBAL MEMORY CONFIGURATIONS
- HEP
- HYPERCUBE (PLANNED)

## PROBLEM

SOFTWARE LAYERS MAKE A "VIRTUAL MACHINE" OUT OF ANY PARALLEL  
HARDWARE

IMPLICATIONS FOR:

- PERFORMANCE
- PROGRAMMING
- CODE PORTABILITY

## PISCES PROJECT

WHAT SORT OF VIRTUAL MACHINE SHOULD THE SOFTWARE PROVIDE?

- EFFECTIVE PERFORMANCE ACROSS A RANGE OF ARCHITECTURES
- EASE OF PROGRAMMING
- PORTABLE APPLICATIONS CODES

## PISCES DESIGN APPROACH

- SUPPORT RESEARCH IN PARALLEL SCIENTIFIC ALGS.
- BASE SEQUENTIAL LANGUAGE
- SIMPLE SET OF EXTENSIONS
- VIRTUAL MACHINE - PRECISELY DEFINED
- SEVERAL GRANULARITIES OF PARALLELISM
- DISTRIBUTED ACCESS TO DATA
- PORTABLE TO A VARIETY OF PARALLEL MACHINES
- COMPLETE VIRTUAL MACHINE ENVIRONMENT
  - SECONDARY STORAGE ACCESS
  - MULTIPLE USERS
  - ERROR HANDLING
  - RELIABILITY

STATUS  
(4/85)

- F77/UNIX/VAX UNIPROCESSOR VERSION - IN USE
  - FORTRAN EXTENSIONS
  - USER INTERFACE
  - ANALYSIS PACKAGE
- APPLICATIONS - RUNNING
  - ITERATIVE SPARSE MATRIX SOLVER
    - PIPELINED DATA FLOW
    - ASYNCHRONOUS OR SYNCHRONOUS
    - BROADCAST OR POINT-TO-POINT
  - IMAGE UNDERSTANDING/AI PROGRAM
- APOLLO WORKSTATION NETWORK IMPL. - RUNNING
  - 10 WORKSTATIONS, 3 DISKS
  - SAME VIRTUAL MACHINE FOR PISCES USER
  - PERFORMANCE STUDIES
- FLEX/32 IMPLEMENTATION - IN DESIGN
  - 17 PES, SEVERAL DISKS
  - SHARED MEMORY - 3 PERFORMANCE LEVELS
- NEW FORMAL MODEL OF PARALLEL COMPUTATION